

# SAMAS: Scalable Architecture for Multi-resolution Agent-Based Simulation\*

Alok Chaturvedi<sup>1</sup>, Jie Chi<sup>1</sup>, Shailendra Mehta<sup>1</sup>, and Daniel Dolk<sup>2</sup>

<sup>1</sup>Purdue University, West Lafayette, IN, 47907, USA  
{alok, chij, mehta}@mgmt.purdue.edu

<sup>2</sup>Naval Postgraduate School, Monterey, CA, 93943, USA  
drdolk@nps.navy.mil

**Abstract.** Large scale agent-based simulation has become the focus of great interest from both academy and industry in recent years. It has been shown an effective tool for understanding a variety of complex systems, such as market economics, war games, and epidemic propagation models. As the systems of interest grow in complexity, it is often desirable to have different categories of artificial agents that execute tasks on different time scales. With the added complexity, the scalability of a simulation environment becomes a crucial measure of its ability in coping with the complexity of the underlying system. In this paper, we present the design of SAMAS, a highly scalable architecture for multi-resolution agent-based simulation. SAMAS is a dynamic data driven application system (DDDAS) that adopts an efficient architecture that allows large number of agents to operate on a wide range of time scales. It uses gossiping, an efficient broadcasting communication model for maintaining the overall consistency of the simulation environment. We demonstrate the effectiveness of this communication model using experimental results obtained from simulation.

## 1 Introduction

Large-scale agent-based simulation has become an effective modeling platform for understanding a variety of highly complex systems and phenomena such as market economics, war games, and epidemic propagation [1]. What is becoming increasingly clear from the deployment of such environments is that they are powerful media for integrating models of widely differing aggregation and granularity. This multi-resolution property of agent-based simulations has already been demonstrated in the spatial dimensions and in the emergence of multi-agent systems, which support a diversity of agents playing different roles and exhibiting different behaviors.

What has been slower in forthcoming is the support of temporal multi-resolution models. Specifically, as systems and their associated models grow in complexity, it becomes increasingly desirable and necessary to have different categories of artificial agents that execute tasks on different time scales. Consider, for example, a simple network in which an agent that simulates network routers needs to operate on the scale of milliseconds while an agent that simulates a mail server, which depends on

---

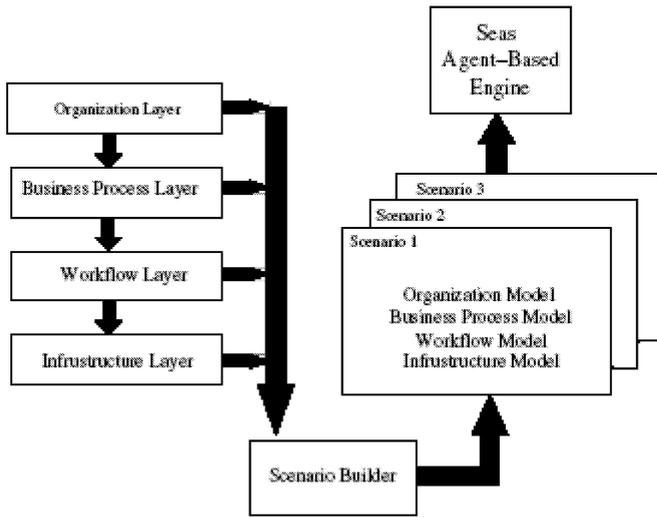
\* This research is funded by the National Science Foundation grant # ACI-0325846.

the routers to get mail, need only operate on the scale of minutes. Such requirements present great challenges to conventional agent-based systems in terms of temporal scalability. These systems typically dictate a single time resolution based on the requirement of the agents running at the highest resolution (lowest granularity, as in the router agents in the above example). Essentially, such systems now run on a single time resolution. Agents are either idle most of the time, causing considerable waste of resources, or they have to be explicitly tracked and activated by the system at different time intervals by the simulation environment, causing excessive overhead for the system. While these strategies are simple to implement, and work well for systems of largely uniform agents, they do not scale to a large number of diverse agents operating on a wide range of time resolutions. In this paper, we propose the design of SAMAS, a dynamic data driven application system (DDDAS) for multi-resolution simulation that uses the gossiping communication model to allow agents to maintain consistency across different time resolutions. Consistency between layers is maintained through a “sample and synchronize” process. In this process, an agent running asynchronously at a higher level can dynamically sample the lower level, which may be another agent (human or artificial), system, component, instrument, and/or sensor, and adjust its parameters.

We approach this problem by first discussing in general terms the power of multi-resolution, agent-based simulation to integrate organizational processes and models at many levels of scope and time (Section 2). We then focus our attention on the problem of temporal multi-resolution and discuss a preliminary design of SAMAS (Section 3.2), which relies upon a gossiping algorithm to coordinate agents operating in different time domains (Section 4). Using simulation, we provide results showing the effectiveness of the gossiping algorithm as a communication model for maintaining global consistency of agents in a simulation environment (Section 5). We summarize our results within the larger context of multi-resolution systems, and briefly present a plan for continuing this venue of research.

## 2 Multi-resolution, Agent-Based Simulation in Support of Organizational Model Integration

Agent-based simulation with robust multi-resolution capabilities offers the potential for coalescing the entire portfolio of processes and models upon which an organization relies for policy and decision-making. Figure 1 shows a conceptual notion of an integrated simulation environment, which takes a layered approach to organizational models. In this framework, models exist at different levels of organization with the Organizational Model being the highest level in terms of generality. The Organizational layer drives the Business Process layer which drives the Workflow layer which, in turn, depends, as all the models do, upon the underlying Infrastructure layer. Each of these layers can be modeled and then solved for different scenarios using the SEAS [2, 3] agent-based simulation engine. The resulting goal is to be able to tie these layers together in an overarching virtual environment, which captures the overall organizational behavior to an acceptable degree of verisimilitude.



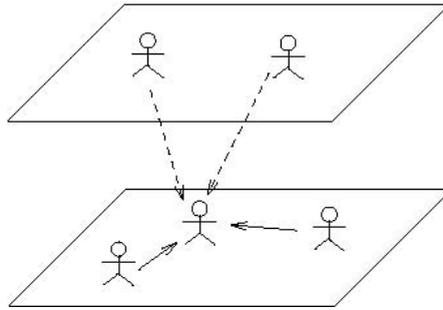
**Fig. 1.** Integration of Multi-resolution Organizational Processes and Models

One of the critical problems in effecting the kind of integration envisioned in Figure 1 is how to handle the time dimension in simulations. The timeframes for organizational processes vary widely in scope. At the strategic level, the focus may be on the next few months or years, at the tactical level days or weeks, and at the operational level, minutes or hours. This disparity in time resolution has a large impact on an organization's modeling processes, which tend to focus model development on a particular domain within a specific timeframe, for example. Although this may be locally efficient, it is often globally suboptimal. The fragmentation of models by domain and timeframe interferes with an organization's ability to integrate and reuse these models, which then can result in the inability to plan and operate to the fullest extent possible. We address the problem of time resolution below by showing an architecture in which agents existing at different time levels in an integrated multi-level model can communicate effectively with one another without consuming inappropriate amounts of overhead.

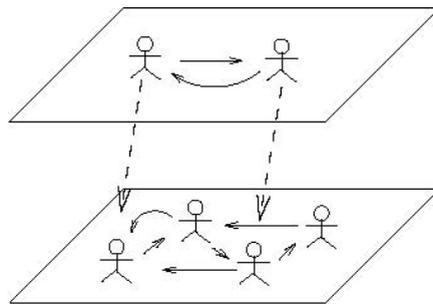
### 3 SAMAS Architecture

#### 3.1 Naïve Design for Multi-resolution Agent-Based Simulation

In most of the complex systems that we study, there are thousands, if not millions, of input elements that may affect the outcome of the system in any number of different ways. Agent-based simulation attempts to capture this complexity by using a large number of artificial agents, each of which plays the role of one or more of the elements in the real system. While existing agent-based simulation systems, given a sufficient number of agents, can effectively mimic the behavior of real world systems, they generally do not effectively take into consideration the temporal relationship



**Fig. 2.** Multi-resolution simulation based on naïve design



**Fig. 3.** Multi-Resolution Agent-based Simulation. Solid lines represent gossiping among agents at the same level. Dashed lines represent sampling among agents at adjacent levels.

among the different agents. As a result, the entire simulation environment uses one global clock time interval for the execution of all agents in the systems. This design, however, does not scale well to large number of heterogeneous agents and may lead to gross inefficiencies in simulation execution. In many sufficiently complex systems, there is often a hierarchy of different types of agents that need to operate on different time scales. The agents that operate at a lower resolution (slower) often have to rely on agents that operate at higher time resolution (faster) for information. To implement such a scenario using the naïve design, we have to assign a central location, e.g. an agent, at the lower resolution level to gather all the information needed by agents at the upper resolution level. The agents at the upper level periodically gather information from the designated party. This scenario is shown in Figure 2. There are two major drawbacks associated with this design:

1. Since there is only one global clock, most of the time the agents on the upper level are doing nothing but checking for information in the lower level. These activities waste the resources in the simulation environment.
2. The agent that is responsible for gathering information at the lower level becomes the single point of contention as the number of agents increase.

### 3.2 SAMAS Design

The design of SAMAS is motivated by the need to build an agent-based simulation system that is capable of scaling up to millions of artificial agents running at multiple time resolutions. The SAMAS simulation environment no longer requires a single uniform global clock. Instead, SAMAS adopts the multi-resolution time interval concept that enables agents to operate at anytime resolution. As shown in Figure 3, the agents on the same level maintain global state information by using gossiping. The gossiping communication model adapted in SAMAS works as the simple steps shown in Figure 4. Essentially, the approach enables the agents at the higher levels to operate asynchronously by sampling the agents at lower level and removes the point of contention in previous design.

Gossiping allows the maintenance of global information without imposing aggressive network bandwidth requirements on the underlying systems. By maintaining the consistent global information among all agents at the same level of time resolution, agents at the higher level can periodically contact any one of the agents at the lower level to obtain necessary information. The agents can safely assume with a high level of confidence that this acquired information, resulting from gossiping communication among agents, truly reflects the actual state information within a bounded degree of error.

- 
1. **Initialize**, for every agent  $i$ , the set  $A_i \leftarrow \{i\}, \forall 0 < i < n$
  2. **Loop** until  $A_i = \{0, 1, \dots, n\}, \forall 0 < i < n$
  3. Agent  $i$  randomly chooses a set agents,  $S, i \notin S, |S| = N_s < n$  and send  $A_i$  to each of them
  4. Agent  $i$  randomly receives a set  $R, |R| \leq N_R < n$  of messages addressed to it, so that  $A_i = A_i \cup A_j, \forall A_j \in R$
  5. **End loop**
- 

**Fig. 4.** Outline of the Gossiping Algorithm Used in Simulation

## 4 Related Work

The gossiping problem, which is also known as all-to-all broadcast, involves each node in a network of  $n$  nodes sending a unique piece of information to the rest of the  $n - 1$  nodes. The gossiping problem has long been the subject of research in various contexts [4]. Researchers investigate a number of different metrics for the solution of the problem. For a complete network of  $n$  nodes, the lower bound on time complexity (the number of communication steps) is established to be  $O(\log n)$  [5] (all log are of base 2 in this paper unless otherwise specified). The minimum number of calls needed for the operation is proved to be  $2n - 4$  for  $n > 4$  [6]. Czumaj et. al.[7] shows the

trade-off relationship between the time and cost of the gossip communication model, where the cost is measured by either the number of calls or the number of links used in the communication. Various strategies have been purposed to perform gossiping on different network architectures with different lower bounds on the performance metrics [8, 9]. In particular, these works show that gossiping can be carried out with desirable performance properties on hypercube architecture. Krumme and Cybenko [10] further show that gossiping requires at least  $\log_{\rho} N$ , where  $\rho$  is the golden ratio, communication steps under the weakest communication model(H1), in which the links are half duplex and a node can only communicate with one other node at each communication step. Gossiping algorithms are at the heart of many fundamental parallel operations, such as matrix transposition and fast fourier transformation [8]. In SAMAS, we implement a simple randomized gossiping algorithm. The details of the operations are discussed in detail in Section 3.2

## 5 Experimental Results

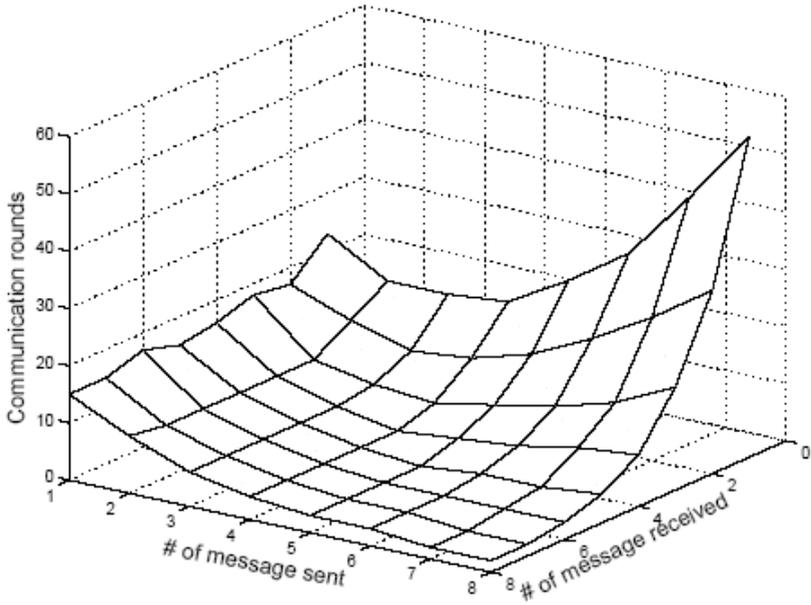
Experiments have been conducted using simulation to measure the performance of the gossip algorithm on a network of agents exhibiting various sizes. We define the following parameters used in the simulation.

1. Number of Communication Rounds is the number of steps it takes to finish the all-to-all broadcast. This is the performance metric measured in the experiments.
2. The number of messages to send (NS) is the upper bound on the number of message an agent can send at each communication round. It represents the outbound bandwidth of an agent.
3. The number of messages to receive (NR) is the upper bound on the number of messages of an agent can receive at each communication round. It represents the inbound bandwidth of an agent.

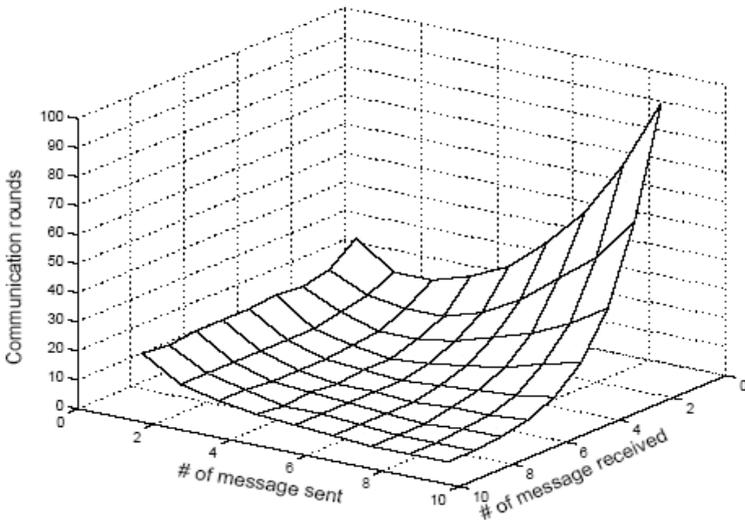
When the number of agent is small, some irregularities are observed (the zigzag in Figure 5). These irregularities are attributed to the randomness nature of the simulation. As the number of agent in the network increase, the run time converges nicely to a continuous surface (Figure 6 and 8).

This performance metric is measured in three different agent network sizes of 128, 256, and 512 agents respectively. We assume overlay networks, which means the networks are fully connected. The simulation is carried out using the procedure shown in Figure 4. We devote the remainder of this section to the discussions and observations of the simulation results.

The number of communication rounds is associated with the run time of the gossiping algorithm. We demonstrate the number of communication round as a function of the outbound bandwidth and the inbound bandwidth of agents in the network (Figures 5 6 and 8). For illustrative purposes, Figures 6 and 7 provide alternative views of the same data in both 3D and 2D. In all three networks with different sizes, we observe the following patterns:



**Fig. 5.** Network size=128. Number of communication rounds as a function of both the number of messages sent and received. Both number of messages sent and received axis are log-scaled



**Fig. 6.** Network size=256. Number of communication rounds as a function of both the number of messages sent and received. Both number of messages sent and received axis are log-scaled

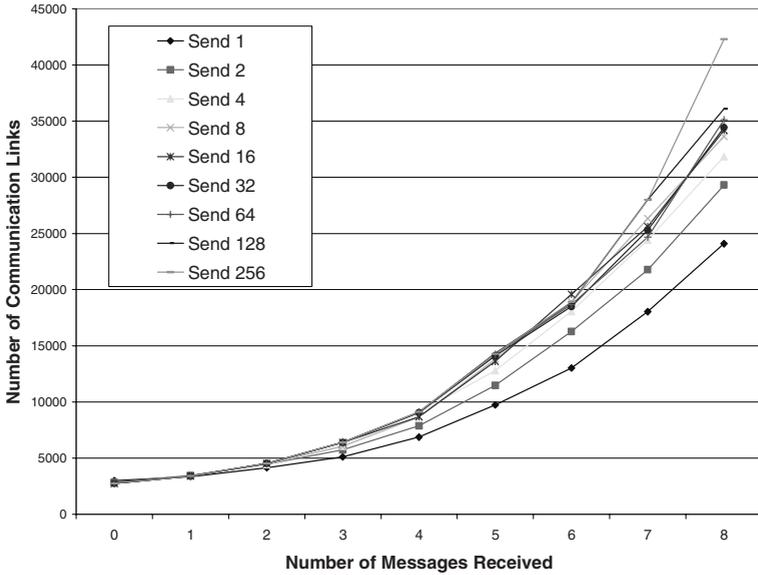


Fig. 7. Two dimensional view of the data shown in Figure 6

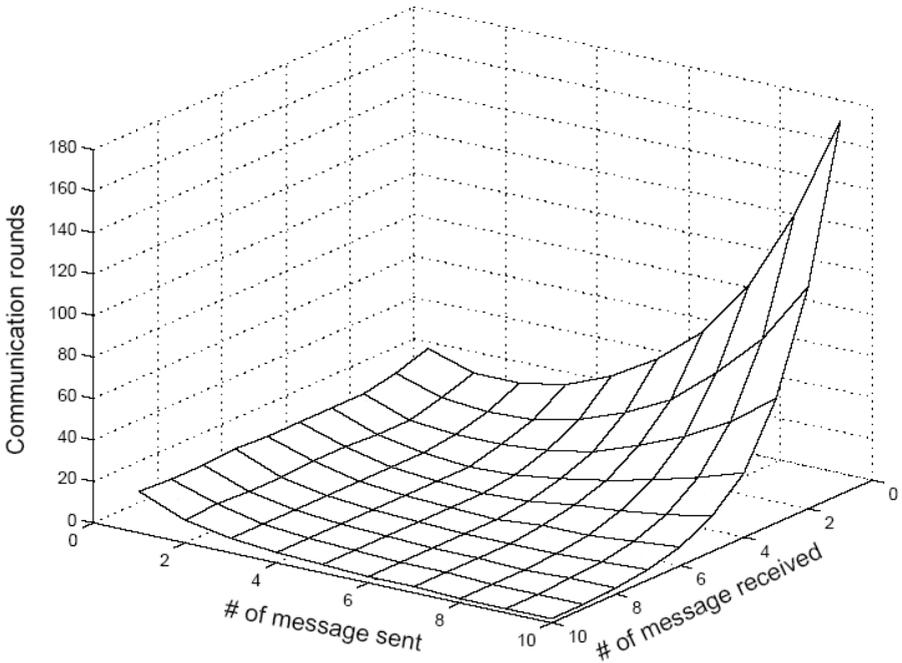


Fig. 8. Network size=512. Number of communication rounds as a function of both the number of messages sent and received. Both number of messages sent and received axis are log-scaled

- Only relatively small network bandwidth is required to perform gossip optimally, or very close to optimal. The optimality referred to here is  $O(\log n)$  steps, where  $n$  is number of agents. In fact, the simulation on average finishes in 7 steps when agents both send and receive 4 messages in each step in  $2^7=128$  agent case, and similarly in other cases.
- A network of agents that has more balanced inbound and outbound bandwidth results in better performance.
- The performance is more sensitive to inbound bandwidth. This asymmetric property can be demonstrated by observing the large increase in time when the agents are allowed to send large number of messages but only receive a small number of them, whereas the inverse is not true.

Our empirical results indicate that the randomized gossiping algorithm adopted in SAMAS can operate at, or very close, to the optimal level shown in previous theoretical works [5,11]. In the context of the multi-resolution simulation model presented in this paper, these results demonstrate that the gossiping algorithm can be effectively used for the purpose of maintaining global state information across agents on the same time interval. For example, if there are 1000 agents of type I operating on a 10 millisecond time interval and several type II agents operating on a 1second time interval, the worst case would require type I agents about 10 steps (100 milliseconds) to update any global information needed by type II agents. This translates into a minimum refresh rate of 10 times/interval for the type II agents, which ensures that type II agents can almost always get the most updated information.

## 6 Conclusion and Future Work

In this paper, we have discussed the importance of multi-resolution simulation for integrating models and processes at all levels of an organization independent of time granularity. We have proposed a preliminary design for SAMAS, a highly scalable dynamic data driven application system (DDDAS) for multi-resolution, agent-based simulation. At the core of SAMAS, we use a gossiping algorithm to efficiently maintain global information among the agents, which operate at different time resolutions. We demonstrate through simulation results that gossiping can be used to implement an architecture that allows large numbers of agents to operate on a wide range of time scales. Should our future experiments with large-scale simulations confirm the feasibility of the SAMAS gossiping approach, we will have taken a significant step in creating a virtual environment for integrating and coordinating mission critical organizational models.

## References

1. Chaturvedi, A., Mehta, S., Drnevich, P., "Live and computational experimentation in bio-terrorism response". In *Dynamic Data Driven Applications Systems*, Kluwer Academic Publishers, (2004)

2. Chaturvedi, Mehta, Dolk, Ayer, "Artificial labor market", *European Journal of Operations Research*, (2003)
3. Chaturvedi, A.R., Choubey, A.K., Roan, J.S., "Active replication and update of content for electronic commerce". *International Journal of Electronic Commerce*, 5, (2003)
4. Hedetniemi, S.M., Hedetniemi, T., Liestman, A.L., "A survey of gossiping and broadcasting in communication networks", *NETWORKS*, 18, (1988), 319–349
5. Bavelas, A. "Communication patterns in task-oriented groups", *J. Acoust. Soc. Amer.*, 22 (1950), 271–282
6. Baker, B., Shostak, R., "Gossips and telephones", *Discrete Mathematics*, 2 (1972), 191–193
7. Czumaj, A., Gasieniec, L., Pelc, A., "Time and cost trade-offs in gossiping", *SIAM Journal on Discrete Mathematics*, 11 (1998), 400–413
8. Grama, A., Kumar, V., Gupta, A., Karypis, G. "Introduction to Parallel Computing, An: Design and Analysis of Algorithms", 2nd Edition. Addison-Wesley (2003)
9. Krumme, D.W., "Fast gossiping for the hypercube", *SIAM J. Comput.* 21 (1992) 365–380
10. Krumme, D.W., Cybenko, G., Venkataraman, K.N. "Gossiping in minimal time", *SIAM J. Comput.*, 21 (1992) 111–139
11. Landau, H. "The distribution of completion times for random communications in a task-oriented group", *Bull. Math. Biophys.*, 16 (1954) 187–201